

**Vysoká škola báňská – Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra informatiky**

**Umělá inteligence pro řízení automobilů**  
**Artificial Intelligence for Self-driving Car**

**2014**

**Radek Slíva**

## Zadání bakalářské práce

Student:

**Radek Slíva**

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

Umělá inteligence pro řízení automobilu  
Artificial Intelligence for Self-driving Car

Zásady pro vypracování:

Umělá inteligence je součástí mnoha systémů, které využíváme v běžném životě. V poslední době se přesunují do oblasti inteligentních systémů dopravy. Tato oblast se nazývá Intelligent Transportation Systems (ITS). Cílem této práce je prostudovat možnosti využití umělé inteligence (výpočetní inteligence) pro řízení automobilů. Práce bude nejprve seznamovat čtenáře s projektem DARPA pro oblast autonomního řízení a s nejspěšnějšími systémy, které výzvu DARPA Grand Challenge přijaly. Dále pak bude pojednávat o existujících simulátorech automobilových závodů, do kterých je možné zapojit vlastní kód umělé inteligence. Praktickou částí je implementace jednoduché umělé inteligence pro řízení automobilu v jednom ze simulátorů nebo v simulátoru vlastním.

1. Prozkoumat možnosti využití umělé inteligence pro řízení automobilu.
2. Nastudovat nejúspěšnější projekty, které se účastnily výzvy DARPA Grand Challenge.
3. Prozkoumat dostupnost simulátorů automobilových závodů.
4. Implementovat jednoduchou umělou inteligenci pro řízení automobilu v jednom ze simulátorů, nebo v simulátoru vlastním.
5. Kriticky zhodnotit vlastní výsledky.

Seznam doporučené odborné literatury:

Sebastian Thrun et al., "Stanley: The Robot that Won the DARPA Grand Challenge", Journal of Field Robotics, Wiley Periodicals, p. 661-692, June 2006, v. 23, no. 9

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Karel Mozdřen**

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014



doc. Dr. Ing. Eduard Sojka  
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

## Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: 6. května 2014

*Radek Zliva*  
.....  
podpis studenta

## **Poděkování**

Rád bych poděkoval panu inženýrovi Karlu Mozdřeňovi za odbornou pomoc a konzultaci při vytváření této bakalářské práce. Také bych chtěl poděkovat mé přítelkyni a rodině za podporu, které se mi dostalo.

## **Abstrakt**

Práce pojednává o řízení automobilů pomocí umělé inteligence. Proto je také jejím úkolem zjistit, jaké se zde vyskytují přínosy a problémy chování při chybovosti ve virtuální formě a také v reálném životě. Dále informuje a popisuje řešení projektů DARPA, které byly úspěšné v oblasti řízení automobilů na předem neznámém povrchu a trati. Bude zde navrženo a popsáno řešení pro pohyb vozidla v situacích jako jsou zatáčky v různých úhlech, vyhýbání se překážkám a rozhodování se při více možných trasách průjezdu do cíle. Budou také popsány existující simulátory, na které půjde aplikovat autonomní řízení. Na závěr popisují úskalí, se kterými se potýkají vývojáři vozidel s umělou inteligencí. Praktická část práce spočívá v implementaci umělé inteligence řízení automobilů v prostředí vlastní implementace simulátoru.

## **Klíčová slova**

simulátor; umělá inteligence; neuronové sítě; autonomní řízení; darpa grand challenge;

## **Abstract**

The work deals with the management of vehicles using artificial intelligence. Therefore, it is also its mission to find out what kind of benefits and behavior problems are occurring in errancy in virtual form and also in real life. Furthermore, informs and describes DARPA projects solutions which have been successful in field of driving cars on unknown surface and track. There will be suggestion and description of solution here for the movement of vehicle in situations such as curves in different angles, avoiding obstructions and decision making at multiple route choices of transit to the final destination. There will be also described the existing simulators, to which will be potentialities to apply autonomous control. Finally, I describe the difficulties faced by vehicle developers with artificial intelligence. The practical part of work consists in the implementation of artificial intelligence for operating cars in an actual implementation environment of the simulator.

## **Key words**

simulator; artificial intelligence; neural networks; autonomous steering; darpa grand challenge;

## Seznam použitých zkratek

<b>Zkratka</b>	<b>Význam</b>
<b>ABS</b>	Protiblokovací systém kol
<b>ASR</b>	Systém regulace prokluzu kol
<b>EU</b>	Evropská Unie
<b>TORCS</b>	The Open Racing Car Simulator
<b>HDP</b>	Hrubý domácí produkt

## Obsah

Úvod.....	- 9 -
1 Umělá inteligence pro řízení automobilu .....	- 10 -
1.1 Adaptivní světlomety .....	- 10 -
1.2 Elektronický stabilizační program.....	- 12 -
1.3 Projekty řízení automobilů DARPA.....	- 13 -
1.3.1 Historie Grand challenge.....	- 13 -
1.3.2 Průběh soutěže.....	- 14 -
1.3.3 Grand challenge 2004.....	- 14 -
1.3.4 Grand challenge 2005.....	- 15 -
1.3.5 Technologie automobilu Stanley.....	- 16 -
1.3.6 Plánování trasy .....	- 16 -
1.3.7 Ovládání vozidla.....	- 17 -
1.4 Automobilové simulátory.....	- 18 -
1.4.1 TORCS .....	- 18 -
1.4.2 Vlastní simulátor .....	- 19 -
1.5 Praktická realizace.....	- 20 -
1.5.1 Analytický algoritmus pro řízení automobilu.....	- 20 -
1.5.2 Implementace umělé inteligence pro řízení vozu .....	- 22 -
1.6 Neuronové sítě.....	- 23 -
1.6.1 Aplikace neuronových sítí.....	- 24 -
1.6.2 Neuron .....	- 24 -
1.6.3 Vícevrstvá dopředná síť .....	- 25 -
1.6.4 Back-Propagation .....	- 26 -
1.7 Genetické algoritmy .....	- 29 -
1.7.1 Genetické učení neuronových sítí .....	- 29 -
Závěr .....	- 32 -
Použitá literatura .....	- 33 -
Seznam příloh.....	I



## Úvod

Jedním z důvodů, proč jsem si vybral téma Umělá inteligence řízení automobilů pro mou bakalářskou práci, je především zájem o automobily a umělou inteligenci. Tyto dva pojmy jsou hlavním tématem mé práce. Oblast je velmi rozsáhlá v automobilové technice, kde dnešní vozy jsou doslova prošípovány těmito systémy, stejně jako v ostatních řídících a robotizovaných systémech. Řízení umělou inteligencí by nejen mělo automobil řídit po dané trase, ale i reagovat na nečekané události a správně vyhodnotit rizika, která mohou nastat. Považuji za důležité napsat práci nejen o řízení automobilů umělou inteligencí, ale také podstatu tohoto oboru v automobilismu a jeho využití v dnešní době a možné uplatnění v budoucnu. V poslední době byl zaznamenán velký pokrok v této oblasti a fascinují mne například projekty DARPA, které jsme zatím znali převážně z televize. Jedna z jejich ukázek je například v dobře známé soutěži DARPA Grand Challenge. Vozidla se pohybovala v terénu po nekvalitních cestách. Tato vozidla by se dala docela dobře označit jako takové pojízdné a výpočetní laboratoře s velkým počtem čidel, senzorů či GPS navigace. Já osobně v této práci popisuji navigaci vozidla ve virtuální podobě a řeším základní problémy umělé inteligence. Autonomní řízení automobilu by mohlo sloužit jako základ pro využití v reálném životě. Motivuje mne myšlenka, kdy lidé, kteří nejsou schopni používat běžný automobil z důvodu tělesného hendikepu, se stanou najednou plně mobilními a ukázkovými účastníky silničního provozu. Tato inteligence se samozřejmě dá rozvinout nejen na automobily, ale i pro komplexní potřeby moderního člověka.

# 1 Umělá inteligence pro řízení automobilu

Umělá inteligence je vědní obor ale také vlastnost. Je využívána pro řešení složitých problémů. Definice umělé inteligence je mnoho, jako nejvystižnější považuji tuto: *Umělá inteligence (jako vlastnost) je tedy souhrn určitých předpokladů a schopností umělého (neživého) objektu, které mu umožní správně reagovat na situace a úlohy, jejichž řešení nemá ve formě algoritmu uloženo v paměti.* (KONEČNÝ, TRENTZ 2010, s. 7)

Umělá nebo také výpočetní inteligence nejen řídí z větší či menší části automobil či jiné motorové stroje, ale dokonce má za úkol řídit dopravu či zavolat pomoc při nehodě. Umělá inteligence je tedy přítomna při řízení automobilu na mnoha místech a nemusíme si toho ani všimnout, jak chytré systémy pracují pro nás a naši bezpečnost především.

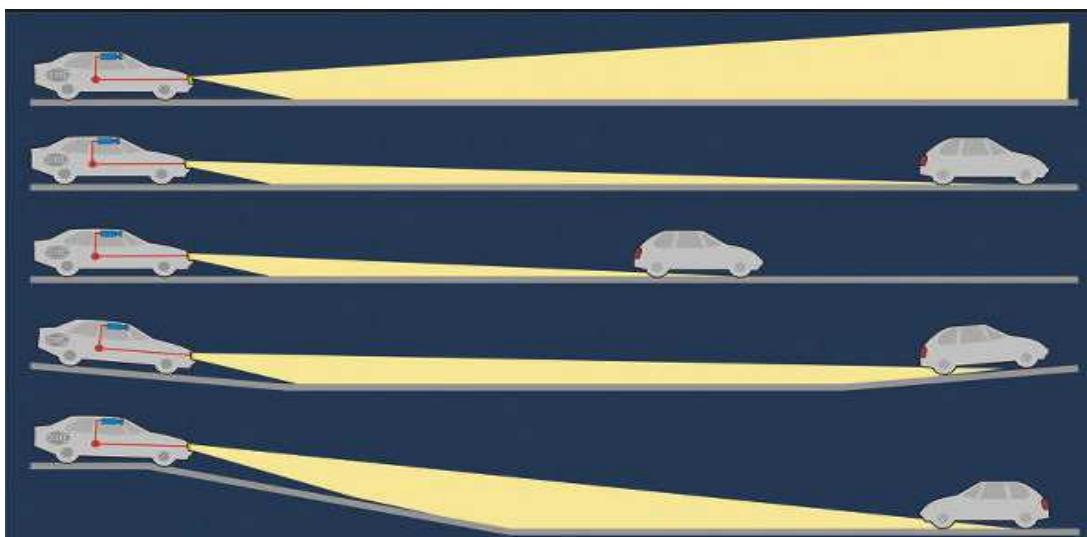
## 1.1 Adaptivní světlomety

Jedná se o inteligenci natáčení světlometů povětšinou bixenonových či LED diodových. Bylo zjištěno dle průzkumu organizace BESIP, že 15% řidičů by díky tomuto systému zabránilo nehodě při průjezdu zatáčkou či horizontu, jelikož by dříve spatřili překážku. Více se lze dočíst zde [4].

Fungování systému adaptivních světlometů je založeno na senzorech či kameře ve zpětném zrcátku. Proto tento systém dokáže rozpoznat blížící se vozidlo v protisměru či ve směru jízdy a také chodce. V ostatních situacích svítí světla na plný výkon. Když se automobil blíží k vozidlu ve stejném směru jízdy, tak světlomety dokážou vyříznout z osvětlené plochy daný automobil tak, aby jezdec v druhém vozidle nebyl oslněn. Toto platí ovšem pro více vozidel, dokonce i pro chodce kráčející na chodníku (obr. 1.1.1). Mezi další vlastnosti patří natáčení světlometů do zatáček, světla se rozhodují v křižovatkách podle natočení volantů. Dále bych zmínil naklopení úhlu reflektorů dle horizontů (obr. 1.1.2), které umožňují stálé osvětlení potřebné plochy pro bezpečné řízení. Systém musí vykazovat inteligenci, aby rozpoznal objekty, kupříkladu ležící pneumatika či zvěř na silnici musí být osvětlena, nikoliv vybrána z osvětlené plochy.



*Obrázek 1.1.1: Výběr osvětlené plochy*  
(<https://www.youtube.com/watch?v=yECf0b5Gfrw>)

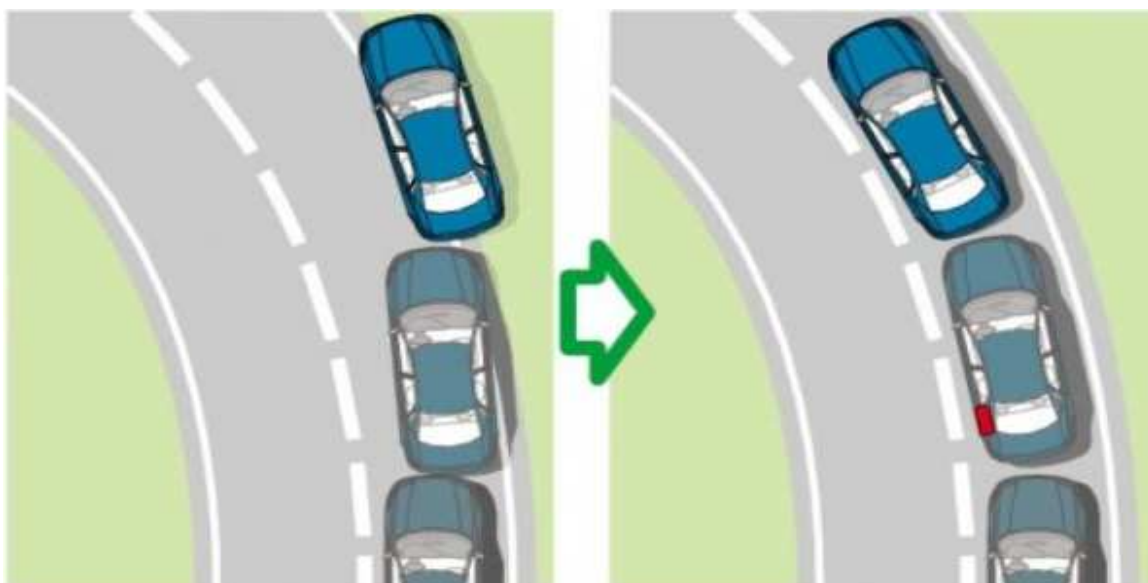


*Obrázek 1.1.2: Úhel světlometů dle horizontů*  
(<http://img.topky.sk/autoviny/big/159146.jpg/technika-adaptive-head-lights-adaptive-ne-svetlomety.jpg>)

Systém adaptivních světlometů je zatím pouhou výsadou prémiových vozidel, přesto rychlý postup vývoje nám možná zajistí větší dostupnost podobných systémů.

## 1.2 Elektronický stabilizační program

Elektronický stabilizační program, značený zkratkou ESP z anglického Electronic Stability Program, je systém, jež se vyvinul z ABS a ASR. Systém je založen na myšlence cíleného zásahu do řízení v kritických situacích. ESP dokáže 30x častěji vyhodnotit situaci stability vozu než řidič. Principem správné funkce ESP jsou dva základní úkony, prvním z nich je, kterým směrem chce řidič, aby auto jelo, a druhým, kde doopravdy směřuje. Proto je automobil s tímto systémem vybaven řadou senzorů, jako například snímačem natočení volantu nebo senzorem plynového pedálu. Nedotáčivost se vyznačuje neochotou zatáčet a zároveň se jedná o typ smyku vozidla. Na takovou situaci ESP zareaguje přibrzděním kola nebo kol obou na vnitřní straně zatáčky. Zároveň sníží výkon motoru tak, aby potlačil špatnou přilnavost pneumatik (obr. 1.2.1).



*Obrázek 1.2.1: Rozdíl v nedotáčivosti s ESP*

*([http://cs.autolexicon.net/obr\\_clanky/cs\\_esp\\_0021-500x282.jpg](http://cs.autolexicon.net/obr_clanky/cs_esp_0021-500x282.jpg))*

Přetáčivost nebo také prudký vyhýbající manévr má za následek přílišné zatočení vozidla. Tento druh smyku je hůře zvládnutelný než nedotáčivost. Je zapotřebí přibrzdit kolo či kola na vnější straně zatáčky (obr. 1.2.2). V ojedinělých případech, když přibrzdění vnějšího kola nepomůže, tak systém vyhodnotí, že je zapotřebí krátkodobě zvýšit akceleraci. Proto má ESP velký podíl v bezpečnosti na silnicích, tudíž EU rozhodla, že všechna vozidla prodaná od roku 2014 musí mít nainstalováno ESP povinně [5].



*Obrázek 1.2.2: Rozdíl v přetáčivosti s ESP*

([http://cs.autolexicon.net/obr\\_clanky/cs\\_esp\\_002a-500x281.jpg](http://cs.autolexicon.net/obr_clanky/cs_esp_002a-500x281.jpg))

### 1.3 Projekty řízení automobilů DARPA

Jedná se o jednu z nejtajemnějších organizací spojených států amerických, která vznikla v roce 1958 jako odpověď vypuštění sputniku Sovětským svazem. Měla za úkol vrátit Ameriku na technologické výsluní ve spolupráci s NASA nebo také měla vybudovat během Studené války Národní komunikační systém. Tehdy se jednalo o agenturu ARPA bez slovíčka Defense (Advanced Research Projects Agency), název platil do roku 1972. Tato organizace spadá pod ministerstvo obrany USA, které je také zodpovědné za vývoj vojenských technologií. Přestože se jedná o vojenský výzkum, už v minulosti se potvrdily případy, kdy se tento vývoj využil v běžném životě. Vznikl zde například systém na zjištění polohy známý jako Transit. Jedná se o obdobu dnešního systému GPS. Už od počátku organizace se jednalo o malou agenturu, která se především zaměřuje na malé projekty, kde jejich dokončení je v horizontu 2 až 4 let. Finanční podporu mají velice štědrou, jelikož pramení z vládních zdrojů. Díky tomu je jejich roční rozpočet okolo 3,2 miliardy dolarů. Také proto jsou financovány i menší vývojové týmy, které pochází z univerzit nebo z malých či větších firem. Tato organizace má okolo 240 zaměstnanců, kde přibližně polovinu z nich tvoří techničtí odborníci. DARPA podporuje výzkumné projekty z mnoha oblastí. Zaznamenala úspěchy například v oblasti biologie, společenských věd, strojírenství, informatiky, chemie, medicíny, fyziky, matematiky a dalších. Více je k dispozici v práci [6].

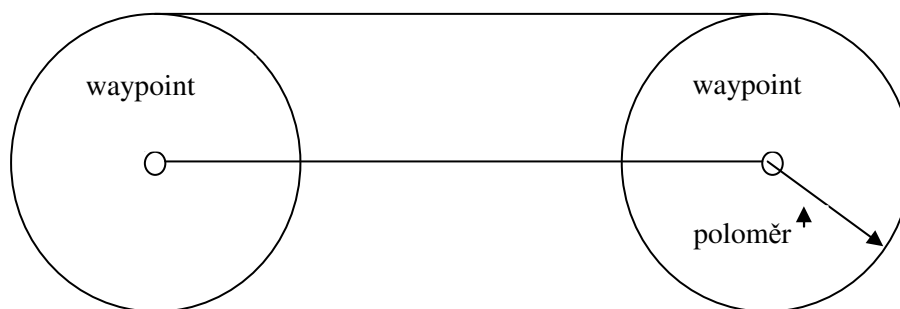
#### 1.3.1 Historie Grand challenge

Mezi nejznámější projekty DARPA patří Grand challenge. Jedná se o soutěž plně autonomních vozidel, která mají za úkol pohybovat se na dlouhé trase v různorodých

podmínkách v co nejkratším čase. První ročník se konal v roce 2004, další následoval o rok později a v roce 2007 se konala soutěž Urban challenge, která byla velice podobná Grand challenge, leč se jednalo o navigaci ve vojenském městečku. Je to soutěž, která byla jako první v tomto okruhu umělé inteligence, kde soutěžní týmy musely připravit své automobily na překonávání velké vzdálenosti bez jakéhokoliv zásahu do řízení. Automobil byl navržen na různorodé tratě (silnice, členitý terén) a měl za úkol se vypořádat s nečekanými překážkami, jako jsou kopce či velké kamení. Tato vozidla se musela dopravit od startu k cíli. Soutěže se zúčastnily týmy z celého světa, byla zde ovšem podmínka, že alespoň jeden člen z týmu musí být občanem Spojených států amerických. Počet účastníků každým rokem stoupal, přičemž odměna činila 1 000 000 amerických dolarů a později v roce 2005 se zvýšila na 2 000 000 amerických dolarů.

### 1.3.2 Průběh soutěže

Dvě hodiny před startem soutěžní týmy dostaly data v elektronické podobě ve formě cd-rom s GPS souřadnicemi, tzv. waypoints s poloměrem, který definuje kruh, kudy musí závodní automobily projet. Tento údaj značí velikost koridoru mezi dvěma waypointy (obr. 1.3.2.1), tudy se autonomní automobily nejen musí pohybovat, ale musí i dodržet časový limit mezi dvěma waypointy, jelikož by ze závodu byly vyřazeny. Mezi další podmínky patří pohyb, při kterém se auto musí dotknout země minimálně každých 5 sekund. Všechna soutěžní vozidla musí být následována dalším (běžným) automobilem s lidskou posádkou, která bude sledovat a kontrolovat dané vozidlo. Také má být opatřeno červeným tlačítkem stop pro nouzové vypnutí. U všech soutěžních týmů platí pravidla silničního provozu pro dodržování rychlosti na komunikacích.



Obrázek 1.3.2.1: Formát trasy

### 1.3.3 Grand challenge 2004

První Grand challenge se konal v USA, 13. března v Mohavské poušti. Více lze dočíst zde [3]. Trať započala nedaleko města Barstow v Kalifornii a vedla do městečka Primm ležícího na hranici s Nevadou (obr. 1.3.3.1). Délka trasy byla přibližně 240 km (150 mil) a vedla skrz mezistátní dálnici Interstate 15. Úspěšnost týmů byla ovšem mizivá, jelikož závod nedokončil ani jeden tým. Nejúspěšnější tým dovedl svůj soutěžní automobil pouhých 11,6 kilometrů, to

činí asi 4,5% celé délky trasy. Vyskytovala se zde totiž řada problémů, se kterými se týmy potýkaly a znemožnily jim tak dokončit trať. Dva soutěžní týmy byly vyřazeny ještě před startem. Dalšímu vozidlu po startu se zablokovaly brzdy. Některá auta neměla dostatek síly pro vyjetí prudkých kopců. Soutěžní vozidlo, které dojelo nejdále, najelo na násyp a při pokusech o vyproštění se díky protáčení kol vznikly pneumatiky. Jednomu automobilu se podařilo vyrazit ze startu opačným směrem a bylo tedy okamžitě diskvalifikováno. Zbytek se buď převrátil v ostrých zatáčkách, nebo se zasekl o překážku a už se nerozjel.



Obrázek 1.3.3.1 :Mapa trasy

([http://media2.s-nbcnews.com/i/msnbc/Components/Maps/Americas/Darpa\\_Grand\\_Challenge\\_v2.gif](http://media2.s-nbcnews.com/i/msnbc/Components/Maps/Americas/Darpa_Grand_Challenge_v2.gif))

#### 1.3.4 Grand challenge 2005

Další ročník se konal o rok později 8. až 10. září. Bylo zapotřebí zdolat 212 kilometrů (132 mil). Cena za první místo byla 2 000 000 dolarů. Tuto trasu dojelo 5 vozidel, z toho 4 v požadovaném čase. Dalších 18 soutěžních automobilů nedojelo do cíle. Vítězem bylo družstvo ze Stanfordské univerzity v Kalifornii s vozidlem Stanley (obr. 1.3.4.1), které zvládlo trasu za 6 hodin a 53 minut.





Obrázek 1.3.4.1: Automobil Stanley [1]

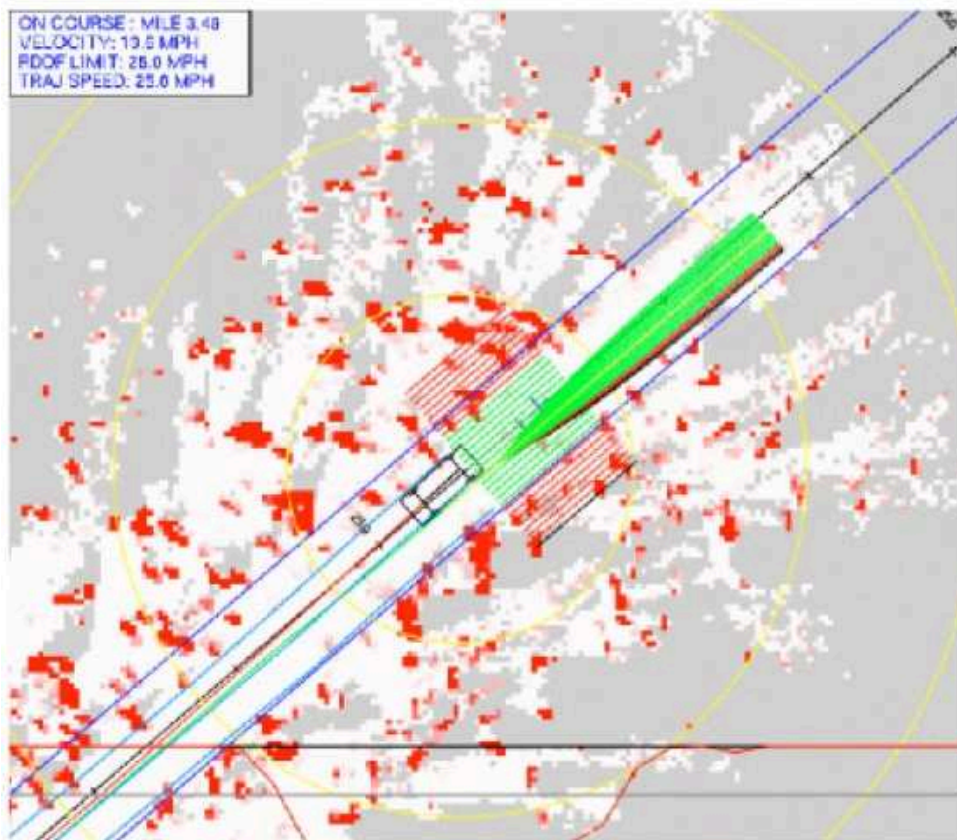
### 1.3.5 Technologie automobilu Stanley

Stanley je postaven na vozidle Touareg s přeplňovaným dieslovým motorem Volkswagen r5 o obsahu 2.5 litrů s pohonem všech čtyř kol. Má automaticky uzavíratelný diferenciál a variabilní nastavování světlé výšky podvozku. Od běžného civilního auta se téměř neliší, když pomineme čidla na střeše. Mozkem automobilu je 6 základních desek s procesorem Pentium M. První dvě desky se starají o kontrolu řízení vozidla. Další deska se využívá k zaznamenání dat a čtvrtá řeší vizualizaci. Zbylé dvě desky byly přidány jako záložní. Proto by měl být Stanley schopen jet i v případě, že by dvě desky nebyly schopny provozu. Dále bylo použito pět dálkových laserových snímačů (LIDAR), které jsou použity pro snímání okolí a především překážek. Těchto 5 měřičů směřuje své měření převážně před vozidlo. Na vozidle je také připevněna kamera zachycující terén vyskytující se před vozidlem a radar měřící delší vzdálenosti. Při určování pozice by se Stanley neobešel bez použití GPS, která má přesnost na 20 centimetrů. Je také nutné vědět o orientaci vozidla, toto zajišťuje GPS kompas. Náklon automobilu je snímán ze všech tří os. Všechny prvky automobilu mají dohromady příkon 500w a jsou napájeny z akumulátoru, který je dobíjen z alternátoru vozidla. Všechny tyto prvky nejsou zpracovávány jedním výpočetním centrem, ale dostávají časová razítka, aby se předešlo zpoždění systému při čekání na požadovaný výsledek.

### 1.3.6 Plánování trasy

Plánování trasy je jeden z dalších důležitých aspektů. Radary ze Stanleyho vytvářejí z přijatých informací 3D objekty a převádí je do 2D mřížky. Na této 2D mapě najdeme bílá pole, která značí sjízdné body, poté červená pole znázorňující naopak nesjízdné trasy, šedá pole jsou pro neznámé oblasti. Jako další krok se vyhodnotí tyto barevné plochy a zelenou barvou se vyznačí bezpečná trasa cesty (obr. 1.3.6.1).





*Obrázek 1.3.6.1: 2D mapa objektů trasy [1]*

### 1.3.7 Ovládání vozidla

Ovládání je jak mechanické, tak elektronické. Mezi elektronicky ovládané prvky patří akcelerace vozu. Ovládání plynového pedálu je přímo přiváděno do řídicí jednotky vozidla. Na stejném principu jsou řízeny brzdy automobilu. Mezi mechanické části patří ovládací ruka, která má lineární pohyb a slouží pro ovládání řadicí páky. Tato mechanická ruka zajišťuje zařazení zpátečky nebo rychlosti pro směr pohybu vpřed. O otáčení volantu se stará elektromotor, který je připojený přímo na tyč řízení vozidla (obr. 1.3.7.1). Více se lze dočíst zde [2].



*Obrázek 1.3.7.1: Elektromotor ovládající volant [2]*

## 1.4 Automobilové simulátory

Každý z nás snad někdy zkoušel automobilový simulátor a jeho řízení. K výběru máme mnoho variant jako například simulátor v běžném provozu, na závodní dráze či ovládaní kamionů. Pak jsou zde simulátory, ve kterých je možné vložit svůj vlastní kód z programovacího jazyka a naimplementovat automobilu inteligenci. Jde tedy o vytvoření virtuálního robota, který bude řídit simulátor sám. Měření sil v těchto simulátorech probíhá prostřednictvím turnajů mnohdy na celosvětové úrovni. Množství simulátorů je velké, ale málo takových, které jsou otevřeny pro volné vkládání kódu.

### 1.4.1 TORCS

Jedná se o populární 3D simulaci automobilových závodů, která patří do linuxové platformy. Tvůrci TORCS se inspirovali starší hrou zvanou Rars. Úkolem programátora tedy je naprogramovat řidiče respektive robota, který bude umět řídit závodní automobil a také se bude umět vyrovnat konkurenci v předem naimplementované počítačové inteligenci. Hra nabízí několik režimů, nejdůležitější však bude režim practice, který je nejvhodnější pro vývoj vlastní umělé inteligence řízení. TORCS podporuje programovací jazyk c a c++. Nabízí 30 různých tratí k dostatečnému procvičení vašeho autonomního jezdce. Nejnáročnější jsou tratě tzv. dirt, které jsou bahnité a plné hrbolů. Z tohoto důvodu musí naimplementovaný systém zvládat různé

změny povrchu. Lze si také vybrat mezi automatickou a manuální převodovkou. Díky tomu je možné si vyzkoušet situace jako v opravdovém voze. Svého robota na řízení lze pak otestovat na nejvyšší úrovni a to při pravidelně konaných turnajích, kterých se pravidelně zúčastňují vývojáři z celého světa. Také je zde možnost vymodelovat své vlastní závodní vozy či okruhy. Více k dispozici na [7].

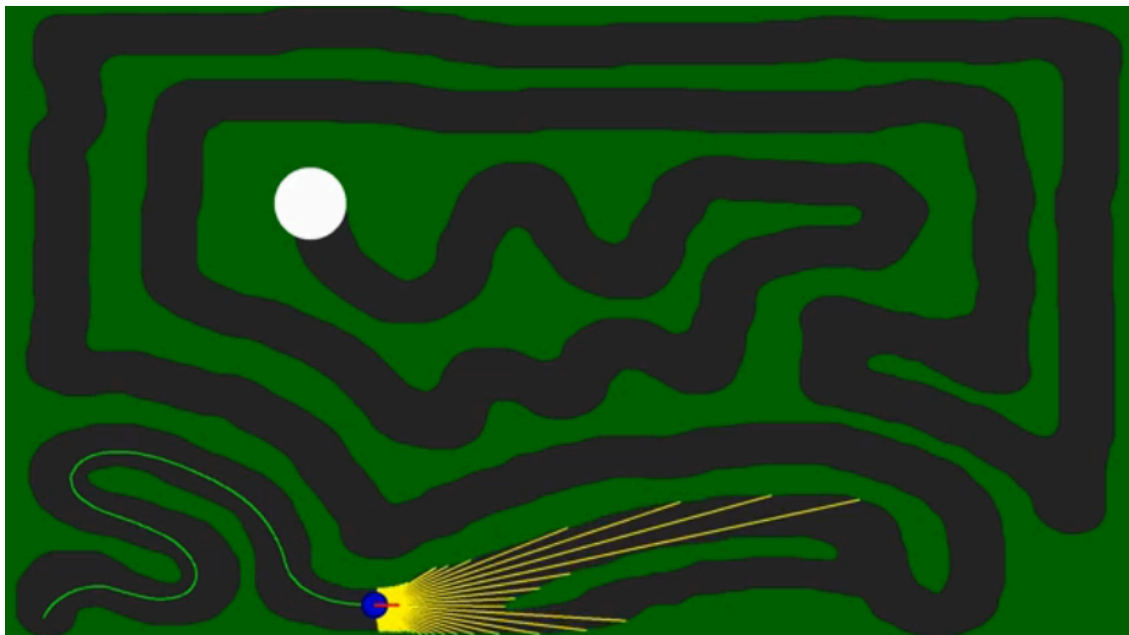
#### 1.4.2 Vlastní simulátor

Rozhodl jsem se pro implementaci vlastního simulátoru, na kterém aplikuji umělou inteligenci. Tento simulátor je zaměřen převážně pro testování naimplementovaného autonomního řidiče pro řízení automobilu po dráze, jelikož vykresluje a vypisuje průběžné výpočetní výsledky a informace získané za běhu. V tomto simulátoru jsou 3 základní požadavky pro úspěšné spuštění. Jedná se o 2 soubory, které se vztahují k jedné dráze. Jako první je mapa dráhy ve formátu png a poté následuje textový soubor. Mapa je vždy černobílá, kdy bílé místa určují možné průjezdní body, a černá barva značí místa mimo trať. Velikost mapy je zpravidla 800x600 bodů, avšak není na tuto velikost omezen. Textový soubor musí uvnitř obsahovat název mapy, souřadnice startu, cíle a velikost poloměru. Tato velikost značí délku poloměru kruhu kolem cílové souřadnice, kde střed této plochy je právě cíl. Při průniku zmíněného obsahu kruhu a vozidla bude vyhodnocen výsledek jako úspěšné splnění dráhy. Dále jsou zde dvě okna programu, po zpuštění jedno vykresluje dráhu, vozidlo a jeho pohyb a druhé vypisuje dodatečné informace při průjezdu tratí. Vozidlo je zde znázorněno jako kruh s červenou linií, která určuje směr pohybu vozidla (obr. 1.4.2.1).



Obrázek 1.4.2.1: Ukázka simulátoru

Simulátor při vykreslování mapy změní barvu z černé na zelenou a z bílé udělá šedou. Pro uživatelsky příjemnější vizualizaci dráhy. Simulátor poskytuje programátorovi signály ke zpracování ve formě snímačů vzdálenosti (obr. 1.4.2.2).



Obrázek 1.4.2.2: Ukázka vizualizace snímačů

## 1.5 Praktická realizace

Tato část pojednává o implementacích vlastních algoritmů, které je možné využít pro problematiku umělé inteligence řízení automobilů. Je zde popsána část pro triviálnější řešení a tím je analytický algoritmus. Dále je tato kapitola zaměřena na implementaci umělé inteligence s využitím genetických algoritmů a neuronových sítí.

### 1.5.1 Analytický algoritmus pro řízení automobilu

Tento způsob vyhledávání cesty patří mezi základní a jednodušší algoritmy pro správné nalezení cesty do cíle. Jedná se o výpočet množiny vektorů a jejich velikostí do jednoho, kterým je vážený průměr těchto vektorů. Tento vektor určuje směr pohybu objektu. Jestliže směrový vektor je na kružnici pod úhlem  $0\pi$ , tak ostatní vektory pro výpočet budou rovnoměrně rozloženy od  $-\frac{\pi}{2}$  do  $\frac{\pi}{2}$ . Já jsem zvolil takových vektorů celkem 180. Kde každému vektoru náleží jeden stupeň na půlkružnici. Tedy od mínus  $90^\circ$  do plus  $90^\circ$ . K výpočtu jednoho takového vektoru musíme zjistit jeho směr a váhu neboli délku. Vzorec na výpočet směru vektoru (přičemž bod A určuje směr pohybu čili  $0\pi$  a B je náš požadovaný bod pro nový směr  $\phi$ ) vypadá následovně:

$$B_x = A_x \cos(\phi) - A_y \sin(\phi) \quad (1.5.1.1)$$

$$B_y = A_x \sin(\phi) + A_y \cos(\phi) \quad (1.5.1.2)$$

Na můj výpočet 180 bodů jsem použil pro  $\phi$  konstantu 0.01745, což reprezentuje jeden stupeň na kružnici. Tedy  $-90^\circ$  bude vypadat takto (přičemž N je  $-90$ ).

$$B_x = A_x \cos(0.01745 \cdot N) - A_y \sin(0.01745 \cdot N) \quad (1.5.1.3)$$

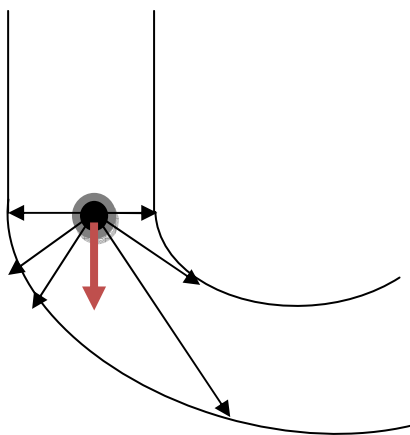
$$B_y = A_x \sin(0.01745 \cdot N) + A_y \cos(0.01745 \cdot N) \quad (1.5.1.4)$$

Po výpočtu všech potřebných vektorů musíme zjistit jejich váhu, respektive délku. Pro výpočet jsem použil vzorec pro vzdálenost dvou bodů přímky (přičemž A je bod aktuální pozice a bod B je průnik přímky s krajnicí dráhy v daném směru).

$$|AB| = \sqrt{(A_x B_x)^2 + (A_y B_y)^2} \quad (1.5.1.5)$$

Váhy jednotlivých vektorů jsou omezeny zdola i shora. Při velké velikosti vah vektorů, které až příliš ovlivňují ostatní, například na dlouhé přímé rovině. Proto je třeba omezit tyto velikosti vzhledem k velikosti tratě a vozu. Další částí algoritmu je výpočet váženého průměru (obr. 1.5.1.1). Využijeme předchozích výpočtů (kde C je náš výsledný vektor pohybu,  $A_1, A_2 \dots A_n$  jsou jednotlivé body na kružnici daných vektorů a  $W_{a1}, W_{a2} \dots W_{an}$  zastupují váhy respektive délky).

$$C = W_a A_1 + W_b A_2 \dots W_n A_n \quad (1.5.1.6)$$



Obrázek 1.5.1.1: Ukázka techniky výpočtu nového vektoru pohybu

Zdrojový kód cyklu pro výpočet váženého průměru ze 180 senzorů pro měření vzdálenosti bude vypadat takto:

```
for (int i=-90;i<91;i++ )
{
    a.x = b.x*cos(i*0.01745) - b.y*sin(i*0.01745);
    a.y = b.x*sin(i*0.01745) + b.y*cos(i*0.01745);
    wa = getProbeDistance(track, this->position, a);
    //GetProbeDistance vrací vzdálenost od okraje vozovky
    r.x += wa*a.x ;
    r.y += wa*a.y ;
}
mag = std::sqrt(r.x*r.x + r.y*r.y);
r.x /= mag;
r.y /= mag;
```

Toto je krok výpočtu, po kterém se virtuální automobil posune o určitou vzdálenost ve směru výsledného vektoru. Takto se postupuje, až vůz protne cílovou čáru.

### 1.5.2 Implementace umělé inteligence pro řízení vozu

Tento algoritmus už využívá neuronové sítě a genetické algoritmy popsané v kapitole 1.6 a 1.7. Program bude učit neuronovou síť za pomoci genetických algoritmů a pohyb virtuálního vozidla po různých trasách. Jedná se tedy o učení neuronových sítí bez učitele. Algoritmus učení bez učitele se hodí na tuto problematiku, jelikož zde budeme hledat jakési maximum a tím bude maximální skóre na dráze. Pro učení zde využijeme poznatky z implementace rozhodovacího algoritmu popsaného výše v kapitole 1.5.1. Nejprve je zapotřebí vytvořit dráhy, které budou sloužit pro učení. Počínaje těmi základními jakou je rovná dráha bez zatáček. Poté s dalšími komplikovanost vzroste tak, aby vůz byl schopen zvládat i opravdu složité tratě s mnohými překážkami. Dále jsou vytvořeny ještě mapy, které mají za úkol ohodnotit výsledek úspěchu jízdy. Čím dále vůz dojede, tím větší bude skóre. Tyto mapy jsou navrženy na jednoduchém principu. Jejich dráha je označena ve stupních šedi a má stejnou trajektorii jako dráha, na které je spuštěna simulace. Start je tedy zcela černý (hodnota 0) a cíl (hodnota 255) má naopak bílou barvu (obr. 1.5.2.1). Poté bude hodnota pixelu, kde vůz dojel, značit skóre úspěchu.



Obrázek 1.5.2.1: Trať pro ohodnocení skóre

Na začátku se načtou všechny mapy společně s údaji, jakými jsou například souřadnice startu, souřadnice cíle a směr, jehož úkolem je natočení vozu na startu. Poté se přejde k trénování. K tomu bude zapotřebí ještě vytvořit neuronovou síť, kterou mám o velikosti [4, 3, 1]. Jedná se tedy o 3 vrstvy, kde v první vrstvě jsou 4 neurony v druhé 3 a nakonec je zde poslední výstupní neuron. Tato topologie je dostačující pro tento typ úkolu. Je to nejpříjemnější kompromis mezi kvalitou výsledku a rychlostí výpočtu algoritmu. Dále se vytvoří populace pro neuronovou síť. Poté můžeme začít s trénováním. Přičemž vstupy jsou v podobě vektorů. Spustí se simulace, která otestuje úspěšnost na každém jedinci. Každý jedinec je testován na trénovací sadě několika tratí a jeho celkové skóre je součtem skóre získaných na všech testovaných tratích. Po každém úspěšném otestování jedné iterace se spustí vizualizace řešení na mapách. Ukládá se nejlepší dosažený výsledek. Účinnost využití umělé inteligence se například prokázala, když vůz narazil na identické zdvojení dráhy, jelikož analytický algoritmus se nedokázal rozhodnout, tak automobil vyjel mimo trať.

## 1.6 Neuronové sítě

Počítačové učení je schopno získávat znalosti pomocí určitého mechanismu na základě zkušeností. Díky schopnosti učení je využitelnost těchto technologií schopna získávat souhrn znalostí za pomoci různých příkladů a tím se zvyšuje využitelnost. Mezi nejrozšířenější postupy pro tvorbu a učení takových systémů patří neuronové sítě s použitím genetických algoritmů.

Neuronové sítě se skládají z neuronů poskládaných do různých topologií. Platí, že výstup neuronu je vstupem dalším neuronům. Počet neuronů v topologii a jejich propojení závisí na složitosti řešeného problému. Z pravidla se určují neurony vstupní, pracovní a výstupní.

### 1.6.1 Aplikace neuronových sítí

Pro úkoly, u kterých je nalezen efektivní způsob řešení, bude s velkou pravděpodobností výhodnější postupovat klasickým způsobem. Jednoduše řečeno neuronové sítě se dají použít všude tam, kde nevádí případná chyba, jelikož tyto sítě jsou úspěšné přibližně na 95%. Mezi úspěšné projekty patří například:

- Syntéza řeči je jedním z prvních úspěšných projektů. Jedná se o neuronovou síť, která je schopná vyslovovat psaný anglický text. Autory jsou Sejnowski a Rosenberg. Nazván byl NETtalk.
- Predikce časových řad je okruh, kde se použití neuronových sítí vyskytuje nejvíce. Převážně jde o předpověď časového vývinu nějaké ekonomické hodnoty. Například cena akcií, HDP či spotřeba elektrické energie.

Ovládání složitých zařízení patří k okruhu úspěšných aplikací neuronové sítě. Především se jedná o ovládání v reálném čase při možné změně prostředí, kde může vzniknout množství nečekaných situací. Jako například řízení robota či ovládání automobilu na komunikacích. Mezi první takové systémy patřil ALVINN (Autonomous Land Vehicle in Neural Networks). Neuronové sítě mají tedy využití v mnoha oblastech, kde selhávají obecná pravidla či vzorce, tam nastupují neuronové sítě.

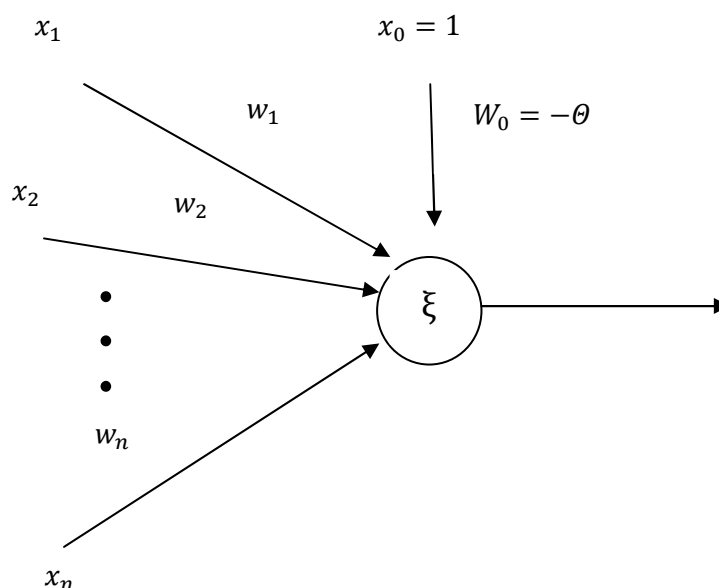
### 1.6.2 Neuron

Jednotkou matematické topologie neuronové sítě je tzv. formální neuron. Ten je převzat a zjednodušen z biologického neuronu. Základní představu neuronu už v roce 1943 navrhli McCulloch a Pitts. Tato představa je stále využívána ve velké části neuronových sítí. Obecně má neuron  $n$  vstupů, které jsou ohodnoceny reálnými váhami, jejíž velikost určuje jakousi propustnost. Propustnost obvykle může nabývat i záporných hodnot. Poté vážená suma těchto vstupních hodnot, respektive vah, je často snižována o práh neuronů neboli bias ( $\theta$ ). Model takového neuronu (obr. 1.6.2.1) lze vyjádřit rovnicí 1.6.2.1

$$y = \sigma \left( \sum_{i=1}^n x_i w_i, \theta \right) \quad (1.6.2.1)$$

V tomto vzorci  $x_n$  reprezentuje vstupy neuronu,  $w_i$  jsou synaptické váhy,  $y$  je výstupní hodnota neuronu. Z toho vztahu bývá  $\sum_{i=1}^n x_i w_i$  uvedeno jako  $\xi$  a jedná se vnitřní potenciál neuronu

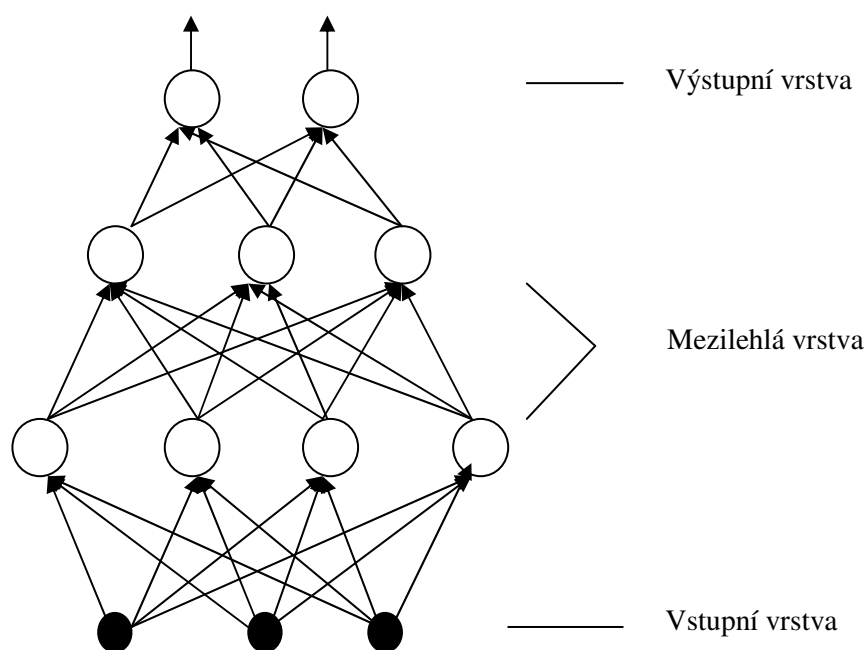




Obrázek 1.6.2.1: Model neuronu

### 1.6.3 Vícevrstvá dopředná síť

Různé typy topologie sítí vznikají díky odlišným propojením neuronů. Dopředná vícevrstvá síť má hlavní funkční vrstvy, kterými jsou vstupní, mezilehlá a výstupní vrstva. Pro tuto topologii platí následující pravidla. Neurony jedné vrstvy se mezi sebou neovlivňují, což znamená neexistenci spojů mezi neurony v jedné vrstvě (obr. 1.6.3.1). Všechny výstupy neuronů v dané vrstvě jsou zároveň vstupy pro všechny neurony vrstvy nadcházející. Platí tedy, že vstupy jednoho neuronu jsou výstupy všech neuronů z předešlé vrstvy. Neurony ve vstupní vrstvě mají výhradně formální funkci a tou je přivést vstupní hodnoty do další vrstvy. Na každý takový vstupní neuron je přiveden pouze jeden vstup. Na rozdíl od vstupní vrstvy řeší mezilehlá vrstva hlavní výpočetní sílu celé sítě. Poslední a tedy výstupní vrstvu chápeme jako vrstvu, která tvoří celý výstup neuronové sítě.

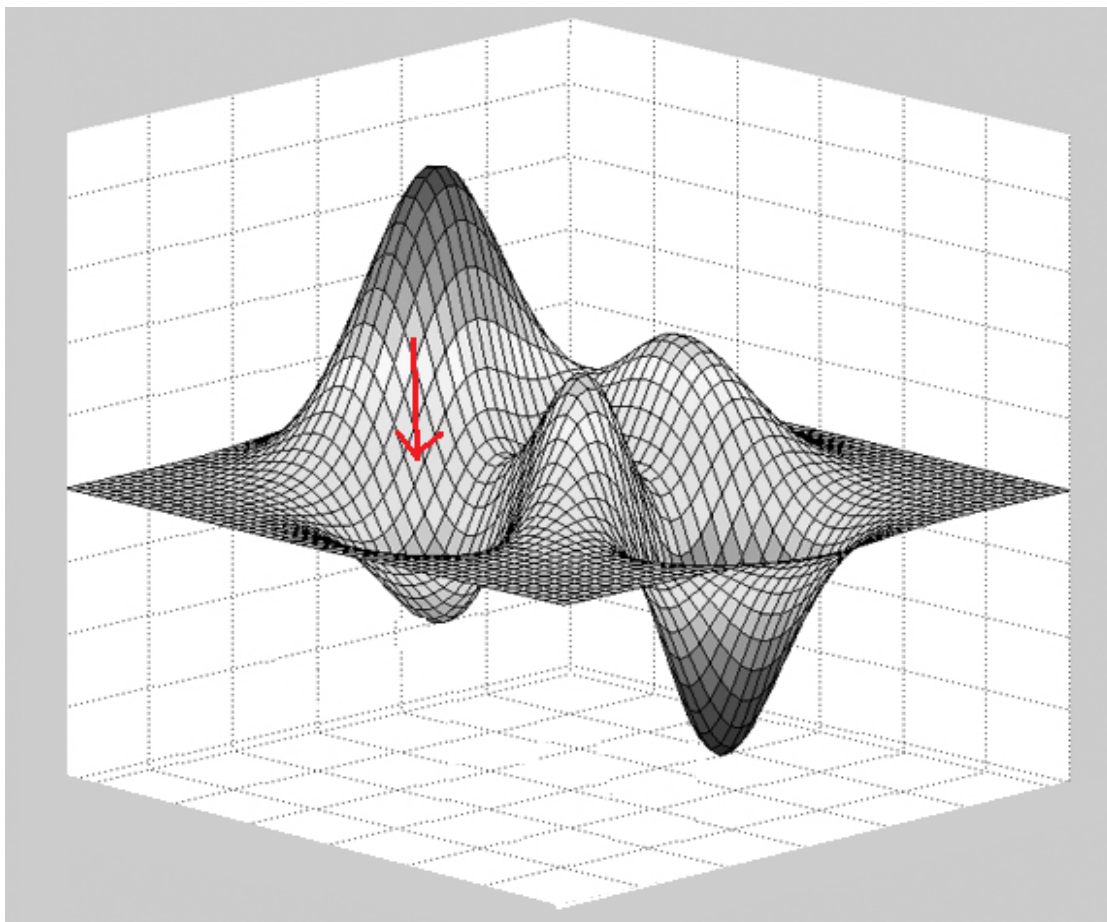


Obrázek 1.6.3.1: Topologie neuronové sítě

#### 1.6.4 Back-Propagation

Jedná se o typ učení neuronové sítě s učitelem. Porovnává vyhodnocené řešení s očekávaným. Tím se zjistí, jak velká je odchylka. Poté se zpětně na základě dané odchylky vytváří velikost změny, o kolik se upraví váhy neuronů tak, aby se odchylka od správného řešení co nejméně lišila.

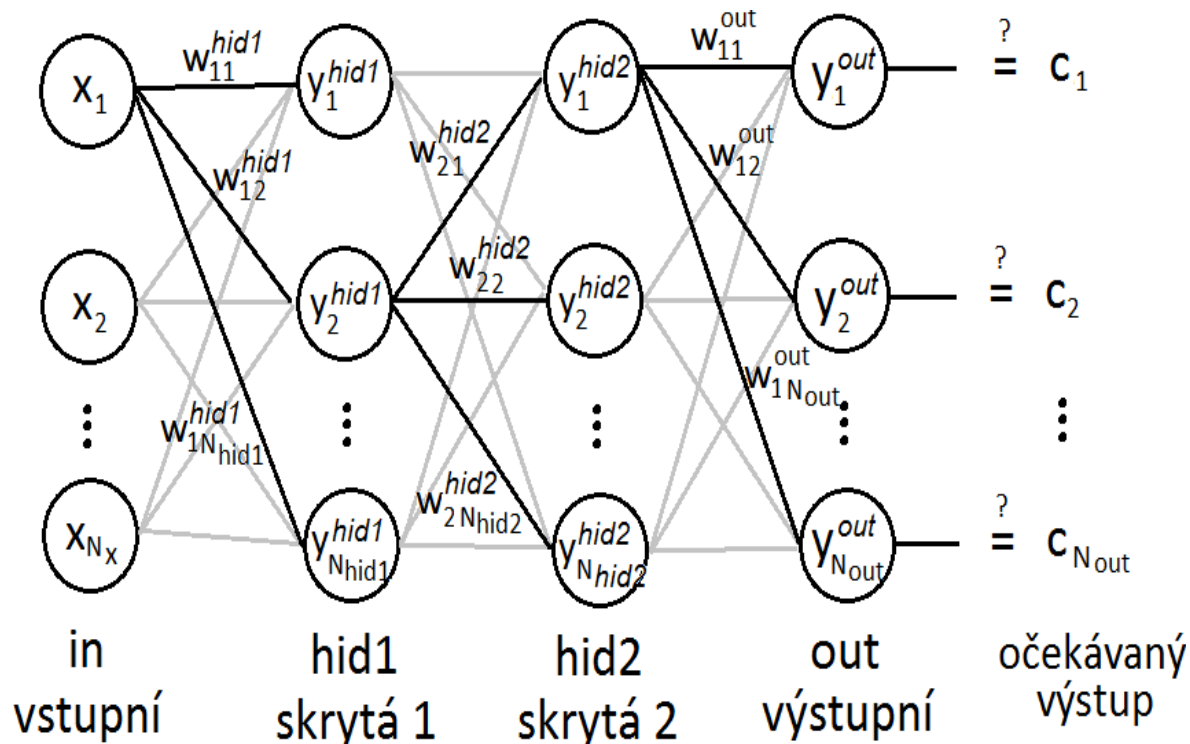
Mějme tedy graf (obr. 1.6.4.1), kde svislá osa značí velikost odchylky neboli chyby. Další osy náležejí parametrům sítě.



*Obrázek 1.6.4.1: Graf s parametry sítě a odchylky*

(<http://www.codeproject.com/KB/dotnet/predictor/locGlobMinim.jpg>)

Proto v grafu je přesně vidět daná chyba pro různé volby parametrů. Cílem je nalézt globální minimum chyby, tedy takové nastavení veličin sítě, aby odchylka byla co nejmenší. Na počátku učení si zvolíme náhodné nastavení parametrů. Je zapotřebí zjistit směr, ve kterém chyba klesá nejstrměji. Hledáme tedy nejstrmější propast. Tento směr je vyznačen červenou šipkou (obr. 1.6.4.1), proto šipka znázorňuje vektor pohybu. Dále postupujeme ve směru vektoru. Chápejme tento vektor jako posun myšleného bodu nastavení neuronové sítě. Postupujeme tak dlouho, až se dostaneme do místa s nejmenší odchylkou. Pro aktualizaci vah v neuronové síti nyní uvažujeme síť s dvěma skrytými vrstvami (obr. 1.6.4.2).



Obrazek 1.6.4.2: Schéma neuronové sítě

Prvky  $x_1 - x_{Nx}$  jsou příznaky na vstupu, díky kterým můžeme klasifikovat. Hodnoty  $y_k^{hid1}$  jsou výstupy individuálního neuronu. A parametry  $w_{ik}^{hid1}$  jsou jednotlivé váhy mezi spojenými neurony. Jako poslední je  $C_{Nout}$ , což je hodnota očekávaného výstupu, která by měla být v nejlepším možném případě shodná s parametrem  $y_k^{hid1}$  jakožto výstup sítě.

Postup vykonávání algoritmu bude tedy následující. Inicializujeme veškeré váhy náhodně v intervalu -1 až 1. Dále vezmeme následující prvek z trénovací sady  $x_1 - x_{Nx}$  a takový, u kterého známe výsledek  $C_{Nout}$  a ponecháme neuronovou síť vyhodnotit vstup, čímž přijdeme na hodnoty  $y_k^{hid1}$ . Nyní vypočteme nové hodnoty vah ze vztahů popsaných níže. Kde znak  $\eta$  značí rychlost učení algoritmu. Jeho hodnota se volí přibližně 0.1. Při nastavení vyšší hodnoty parametru  $\eta$  se učení zrychlí, nýbrž bude to na úkor kvality.

$$\delta_k^{out} = \gamma y_k^{out} (1 - y_k^{out}) (C_k - y_k^{out}) \quad (1.6.4.1)$$

$$\Delta W_{ik}^{out} = \eta \delta_k^{out} y_i^{hid2} \quad (1.6.4.2)$$

$$\delta_k^{hid2} = \gamma y_k^{hid2} (1 - y_k^{hid2}) \left( \sum_{j=1}^{Nout} \delta_k^{out} w_{kj}^{out} \right) \quad (1.6.4.3)$$

$$\Delta W_{ik}^{hid2} = \eta \delta_k^{hid2} Y_i^{hid1} \quad (1.6.4.4)$$

$$\delta_k^{hid1} = \gamma Y_k^{hid1} (1 - Y_k^{hid1}) \left( \sum_{j=1}^{N_{hid2}} \delta_k^{hid2} w_{kj}^{hid2} \right) \quad (1.6.4.5)$$

$$\Delta W_{ik}^{hid2} = \eta \delta_k^{hid2} X_i \quad (1.6.4.6)$$

$$W_{ik}^{out} \leftarrow W_{ik}^{out} + \Delta W_{ik}^{out} \quad (1.6.4.7)$$

$$W_{ik}^{hid2} \leftarrow W_{ik}^{hid2} + \Delta W_{ik}^{hid2} \quad (1.6.4.8)$$

$$W_{ik}^{hid1} \leftarrow W_{ik}^{hid1} + \Delta W_{ik}^{hid1} \quad (1.6.4.9)$$

Dále parametr  $Y$  značí hodnotu funkce Sigmoid neboli aktivační funkci. Velikost se volí většinou 1.0 a jedná se o strmost aktivační funkce. Tímto postupem můžeme naimplementovat učení neuronové sítě pomocí Back-Propagation. To by ovšem v našem případě představovalo přípravu trénovacích dat, které pokrývají pokud možno co největší množství možných vstupů, což není v krátké době realizovatelné. Rozhodli jsme se proto pro trénování neuronových sítí, metodou bez učitele. Jmenovitě jsme pak použili genetický algoritmus pro její trénování. Pro který je nutné pouze vyhodnotit úspěšnost specifického nastavení sítě (vah) a přirozenou evoluci tak získat optimální síť. Genetický algoritmus podrobněji popisují v následující kapitole.

## 1.7 Genetické algoritmy

Je zde popsán postup, který se snaží aplikací principů evoluční biologie nalézt řešení složitých problémů, pro které neexistuje použitelný exaktní algoritmus. Podrobný popis prvků, které se budou zde využívat a také procedury použité pro správné řešení této problematiky.

### 1.7.1 Genetické učení neuronových sítí

Základním prvkem je populace, která je tvořena z generací jedinců, jež se dle předem daného systému vyvíjejí z generace předcházející. Je zde použita terminologie známá z biologie (tab. 1.7.1.1). Princip je tedy jednoduchý, jde o to vybrat nejlepšího jedince, který se hodí pro řešení dané úlohy. Na začátku je tedy náhodně vygenerovaná množina jedinců. Tato množina je posléze otestována pro daný problém a ohodnocena. Následující generace je vytvořena z předchozí populace s pomocí mutace, křížení a selekce. Nejprve se vytvoří dvojice z předchozí populace a poté se aplikuje křížení a mutace. Jednotlivé principy jsou popsány níže.

- Selekce: Čím větší má jedinec skóre, tím větší má šanci na vybrání do páru. Tímto způsobem je větší pravděpodobnost, že v páru budou dva silní jedinci.

- Mutace: S malou pravděpodobností náhodně vybraných chromozomů se změní náhodně velikost genu.
- Křížení: Existuje řada technik křížení, ale vždy jde o jakousi záměnu určité části chromozomu s partnerem. Tedy jedinec zdědí po rodičích určitou část genetické výbavy.

Na takto vytvořených párech se opět spustí simulace úkolu a vyhodnotí skóre. Postupuje se tímto způsobem až do výstupní vrstvy. Více se lze dočíst zde [8].

Tabulka: 1.7.1.1: Rozdíl mezi biologickou (vlevo) a matematickou definicí (vpravo)

Populace	
Množina generací.	Uspořádaná množina generací dle postupného vývoje.
Generace	
Jedná se o množinu jedinců.	Množina jedinců, na níž je prováděna selekce pro případnou novou generaci.
Jedinec	
Jedinec neboli nositel genetického kódu.	Reprezentuje vektor genotypu.
Genotyp	
Genetický popis.	Komplexní vektor, který reprezentuje řešený problém.
Gen	
Elementární zástupce genetické informace.	Zástupce jednoho znaku či symbolu v chromozomu.
Fenotyp	
Fyzický popis genotypu. Například je-li chromozom XX (genotyp), potom to znamená, že člověk je žena (fenotyp).	Výsledná hodnota transformace kódování. Například binárně 101 (genotyp) je dekadický 5 (fenotyp).
Chromozom	
Součást molekuly DNA.	Jeden či více dílů z výsledného vektoru (čísla, symbolu).

Alela	
Konkrétní forma genu.	Množina hodnot, kterých může gen nabývat.
Fitness	
Šance na přežití jedince.	Skóre jedince v řešeném problému.

## **Závěr**

Cílem bakalářské práce bylo podrobněji popsat úskalí a výhody při využití umělé inteligence pro řízení automobilů, zároveň uvést využití takových systémů v praxi a poukázat na úspěšné projekty s danou problematikou a doporučit a vysvětlit správné řešení pro implementaci.

Na základě této analýzy umělé inteligence a s ní spojenou implementací pro řízení automobilů se ukázalo jako klíčové použít neuronové sítě s kombinací genetických algoritmů. Triviální rozhodovací algoritmus byl účinný pro jednodušší dráhy, ale při neočekávaných překážkách či různorodé trati, se ukázala výhoda umělé inteligence a její postupné učení. S vyšším stupněm učení rostla i kvalita trajektorie automobilu po dráze, tudíž se vůz pohyboval plynule a bez zbytečných zaváhání při průjezdu dráhy. Tím se ušetřil i čas dojezdu do cíle.

Výsledkem praktické části je vlastní funkční simulátor s využitím umělé inteligence, který plní dané úkoly a vykresluje postupný průběh učení. Ukládá a zobrazuje maximální dosažené výsledky při pokusech o splnění daných tratí. Tato metodika řešení byla použita v několika úspěšných projektech.



## Použitá literatura

- [1] Stanley: *The Robot that Won the DARPA Grand Challenge*. 2006. Dostupné z: <http://robots.stanford.edu/papers/thrun.stanley05.pdf>.
- [2] TRNĚNÁ, Michaela. *DARPA Grand Challenge: Závod plně automatizovaných vozidel*. 2012. 342090. Referát. Masarykova univerzita. ....
- [3] KAREŠ, Ing. Rostislav. DARPA Urban Challenge 2007 – velký den pro robotiku. [online]. 2008 [cit. 2014-04-30]. Dostupné z: [http://www.odbornecasopisy.cz/index.php?id\\_document=36992](http://www.odbornecasopisy.cz/index.php?id_document=36992)
- [4] Adaptivní světlomety: AFL ( Adaptive Forward Lighting) - Adaptivní světlomety. [online]. [cit. 2014-04-30]. Dostupné z: <http://www.ibesip.cz/cz/ridic/bezpecne-vozidlo/moderni-technologie-vozidel/aktivni-bezpecnost-prvky-aktivni-bezpecnosti/adaptivni-svetlomety>
- [5] MIHÁLIK, Miro. ESP je od listopadu povinné. Bylo to opravdu nutné?. [online]. 2011 [cit. 2014-04-30].
- [6] DARPA. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 29. dubna 2014 [cit. 2014-04-30]. Dostupné z: <http://en.wikipedia.org/wiki/DARPA>
- [7] JANÍČEK, Martin. Cube, Torcs, Trigger. [online]. 2005 [cit. 2014-04-30]. Dostupné z: <http://www.linuxexpres.cz/hry/cube-torcs-trigger>
- [8] BISKUP, Mgr. Roman. *Možnosti neuronových sítí*. 2009. Disertační. Česká zemědělská univerzita v Praze.
- [9] KONEČNÝ, Vladimír a Oldřich TRENZ. *Základy umělé inteligence*. Brno, 2010. MENDELOVA UNIVERZITA V BRNĚ Fakulta provozně ekonomická.

---

## Seznam příloh

Příloha A:      *Vizualizace praktické realizace.....II*

Součástí bakalářské práce je CD.



Obrázek A.1: Vizualizace řešení



Obrázek A.2: Vizualizace řešení



*Obrázek A.3: Vizualizace řešení*



*Obrázek A.4: Vizualizace řešení*



*Obrázek A.5: Vizualizace řešení*



*Obrázek A.6: Vizualizace řešení*